ON-LINE MIXTURE-BASED ALTERNATIVE TO LOGISTIC REGRESSION

I. Nagy[*], *E. Suzdaleva*[†]

Abstract: The paper deals with a problem of modeling discrete variables depending on continuous variables. This problem is known as the logistic regression estimated by numerical methods. The paper approaches the problem via the recursive Bayesian estimation of mixture models with the purpose of exploring a possibility of constructing the continuous data dependent switching model that should be estimated on-line. Here the model of the discrete variable dependent on continuous data is represented as the model of the mixture pointer dependent on data from mixture components via their parameters, which switch according to the activity of the components. On-line estimation of the data dependent pointer model has a great potential for tasks of clustering and classification. The specific subproblems include (i) the model parameter estimation both of the pointer and of the components obtained during the learning phase, and (ii) prediction of the pointer value during the testing phase. These two phases can be joined together in the case of necessity. A real-data experimental comparison with theoretical counterparts shows a competitiveness of the approach in the discussed field.

Key words: *on-line modeling, on-line logistic regression, recursive mixture estimation, data dependent pointer*

Received: January 15, 2016 **DOI:** 10.14311/NNW.2016.26.024
Revised and accepted: October 3, 2016

1. Introduction

The presented paper deals with a problem of modeling discrete variables depending on continuous variables. This problem is generally known as the logistic regression [16]. Classification based on the logistic regression is widely applied in various fields. Only to enumerate, in medicine applications data observed on a patient (e.g., weight, blood pressure, cholesterol level, sex, age, results of various blood tests, etc.) can be analyzed by the logistic regression to obtain a probability of the certain disease and subsequently to classify the patient's state [2, 8, 9, 20]. In

[*] Ivan Nagy, Department of Signal Processing, The Institute of Information Theory and Automation of the Czech Academy of Sciences, Pod vodárenskou věží 4, 18208 Prague, Czech Republic, and Faculty of Transportation Sciences, Czech Technical University, Na Florenci 25, 11000 Prague, Czech Republic, E-mail: `nagy@utia.cas.cz`

[†] Evgenia Suzdaleva – Corresponding author, Department of Signal Processing, The Institute of Information Theory and Automation of the Czech Academy of Sciences, Pod vodárenskou věží 4, 18208 Prague, Czech Republic, E-mail: `suzdalev@utia.cas.cz`

marketing, the logistic regression is applied to predict a customer acceptance before introducing a new product [37], to classify the customer satisfaction by services [24], etc. In social sciences it is used for prediction of the election results, see e.g., [32]. These are only several examples. The application of the logistic regression is not limited by the mentioned areas.

Parameters of the logistic regression are usually estimated by the maximum likelihood estimation [27]. The closed-form solution for maximizing the likelihood function in the case of the logistic regression parameter estimation cannot be found. This causes necessity to use numerical optimization methods, see e.g., [21, 28, 39]. It leads to a series of limitations concerned with the convergence of the algorithms, which means that if the numerical optimizer cannot find the appropriate solution, the convergence fails. Various reasons can lead to nonconvergence, for instance, multicollinearity [27], sparseness [27], complete and quasi-complete separation [1], etc. The computation time of numerical methods depends on the speed of convergence of the algorithm and on the initial conditions, which means it is not fixed. Therefore, it is not guaranteed in advance whether the computations will be ready in time or not. In practical applications, this question may be crucial, especially in the case of working with a very short sampling period.

This paper approaches the problem from a different point of view. The aim of the presented research is to find a possibility to model on-line a switching of working modes of a multimodal stochastic system in dependence on measured continuous data. The system is described by a mixture model, which is estimated under Bayesian framework. Mixture based approaches are intensively developed and applied in various fields [33,41–44,46]. In the context of the paper, the mixture model consists of components describing the working modes of the observed system and the discrete random variable called the pointer, which indicates the currently active component [18]. Thus the model of switching expresses the dependence of the discrete pointer on measurements produced by the components. In the case of continuous measurements the switching model (either constant or the Markov model) takes a form of the logistic regression. As it should be estimated on-line, this is a motivation to search for an alternative solution avoiding the numerical off-line computations and being oriented at analytical solutions.

Bayesian inference offers a series of Markov Chain Monte Carlo (MCMC) methods based on various approximation schemes using e.g., the Student's t-distribution [4, 13, 22]. Other possibilities are given by using the Laplace approximation [25], the probit regression instead of the logistic [22,23], a latent variable model [10,35], the Metropolis–Hastings algorithm [6], etc.

However, the aim of the paper is not to approximate the posterior distribution by sampling methods. The purpose is to explore a possibility of construction of the continuous data dependent switching model that should be recursively (on-line) estimated. This might be a significant contribution for situations, when the model of the discrete variable should be periodically learned from the newly arriving data. Moreover, the considered on-line modeling is also expected to be essential in the case of missing data, i.e., when the discrete variable is measured with some longer period, but should be estimated between them.

The area of the mixture estimation (not concerned with the logistic regression) provides a great amount of published solutions. Papers dealing with this issue are mostly based on (i) the iterative expectation-maximization (EM) algorithm [15], see, e.g., [3, 7, 31, 40, 45]; (ii) the Variational Bayes (VB) approach [14, 26, 36]; (iii) MCMC techniques, e.g., [5, 11].

Unlike them, recursive algorithms of Bayesian identification of linear regression models [34], categorical models [17] and mixtures [18] are directed at using analytical solutions as far as possible and on-line computing, i.e., avoiding numerical computations. They create a basis for the approach presented in this paper.

Here the model of the discrete variable dependent on continuous data is represented as the model of the mixture pointer dependent on data from mixture components via their parameters, which switch according to the activity of the components. To focus on this the most problematic part of the on-line modeling, normal components are demonstrated. Extension up to mixed continuous and discrete data leads in this case to addition of a categorical component with the reproducible Dirichlet statistics, see [17], and will not cause a computational complexity. Moreover, different components (e.g., state-space, exponential) can be also covered, which is planned to be published elsewhere (this will enable fitting different types of data too). On-line estimation of the data dependent pointer model has a great potential for tasks of clustering and classification. This paper indicates a chance of modeling the continuous data dependent pointer. The main contribution of the paper is an alternative to the logistic regression task based on the recursive mixture estimation.

The approach is presented so that to have the separate learning and the testing phases. Within the considered context, the specific subproblems include (i) the model parameter estimation both of the pointer and of the components obtained during the learning phase (based on [17, 18, 34]), and (ii) prediction of the pointer value during the testing phase. These two phases can be joined together in the case of necessity. A real-data experimental comparison with theoretical counterparts shows a competitiveness of the approach in the discussed field.

The remainder of the paper is organized in the following way. Section 2 formulates the problem. Section 3 introduces the used models and provides a theoretical background necessary for understanding the text. Section 4 is the main emphasis of the paper. It presents two alternatives to the multinomial logistic regression based on the recursive mixture estimation. One of them includes the off-line learning phase and the on-line testing phase, which enable to use the accumulated statistics for classifying the data. The second one joins these phases and has them both on-line with updating the statistics based on the newly arriving data. The section explains the approach in details and provides two structural algorithms. Section 5 demonstrates a simple example with simulated data and results of experiments with real data measured on a driven vehicle, where the gear selection is modeled as the discrete variable depending on several driving-related variables. Conclusions and plans of a future work can be found in Section 6.

2. Problem formulation

Let us consider a system which generates values of the discrete random variable $y_t \in \{1, 2, \ldots, K\}$ with K possible values at discrete time instants $t = 1, 2, \ldots, T$.

The system also produces the data vector $\mathbf{x}_t$ of the dimension N, whose entries are continuous random variables observed at time $t = 1, 2, \ldots$.

The problem is formulated as follows:

- based on the available data up to the time T (i.e., when values of y_t are measured) describe the relationships between the dependent variable y_t and the explanatory variable $\mathbf{x}_t$ by a suitable model and estimate its parameters (this is called *the learning phase* of the algorithm), and
- estimate the value of y_t for the newly measured explanatory variable $\mathbf{x}_t$ for the time $t > T$, i.e., classify the data $\mathbf{x}_t$, when values of y_t are no longer measured (*the testing phase* of the algorithm).

In this paper, a mixture of static models – *components* is chosen to describe the relationships between y_t and $\mathbf{x}_t$. The discrete variable y_t plays a role of the *pointer*, which at time t indicates the active component. The data vector $\mathbf{x}_t$ is modeled by each of the involved components. Thus, the logistic regression problem is going to be solved using the recursive estimation of a mixture model.

3. Preliminaries

3.1 Models

In this paper, a mixture model describing the considered system consists of K static components and a model of switching their activities. The components have the form of the following probability density function (pdf)

$$f\left(\mathbf{x}_t|\Theta_i\right), \tag{1}$$

where $i \in \{1, 2, \ldots, K\}$, and Θ_i are parameters of the i-th component. Here each i-th component is represented by the static regression model with the normal noise, i.e.,

$$\mathcal{N}_{x_t(i)}\left(\theta_i, \mathbf{R}_i\right), \tag{2}$$

where $\mathcal{N}_{x_t(i)}$ denotes the normal distribution of the i-th component generating the data $\mathbf{x}_t$, and θ_i is the vector of regression coefficients (the expectation corresponding to the center of the i-th component). The normally distributed noise of the i-th component has the zero mean vector and the covariance matrix $\mathbf{R}_i$. In this way, each component is a Gaussian "hill" positioned in its center (the expectation) and formed by its covariance matrix, all in the multivariate data space of all realizations of the modeled variable. Parameters θ_i and $\mathbf{R}_i$ compose Θ_i, and $\Theta \equiv \{\Theta_i\}_{i=1}^{K}$ is a collection of all parameters of all components.

Switching the active components generating the data $\mathbf{x}_t$ is described by the following pdf (both the probability density function and the probability function are replaced by the abbreviation pdf in the text):

$$f\left(y_t = i|\alpha\right) \tag{3}$$

given by the static transition table

y_t	$y_t = 1$	$y_t = 2$	$\cdots$	$y_t = K$
$f(y_t = i\|\alpha)$	α_1	α_2	$\cdots$	α_K

where a value of the discrete variable y_t points to the component, which generates the data $\mathbf{x}_t$ at time t, and α is the parameter of the pointer model. It is the vector of the dimension K, which contains stationary probabilities α_i of the activity of individual components.

Within the context of the above mixture model, the relationships between the discrete dependent variable y_t and the explanatory variable $\mathbf{x}_t$ are assumed as follows. For each possible value of y_t there exist different (or partially overlapped) data areas in the whole data space of measurements $\mathbf{x}_t$. The data areas are modeled by the mixture components (1) changing their activity according to (3), which both have unknown parameters Θ and α respectively. This covers the first part of the problem formulated in Section 2.

As regards its second part, it requires to estimate the value of y_t using the data item $\mathbf{x}_t$ measured at time t and produced from a certain (but unknown) component. Thus, for each data item at time t it is necessary to determine a probability that it belongs to the i-th component. This gives the classification of the data.

Thus, it can be seen that the formulated subproblems are specified to the estimation of the parameters Θ and α and the pointer y_t in the testing phase. For better understanding the subsequent text, existing Bayesian recursive algorithms of estimating the parameters of individual models (1), or precisely (2), and (3) are recalled below. Both of them are based on the Bayes rule, see e.g., [12, 19].

3.2 Recursive estimation of individual models

In the case of the recursive estimation of the individual model (1) the posterior pdf of the parameter Θ (omitting here the subscript i for the sake of simplicity) is evolved in time in the following way:

$$f(\Theta|x(t)) \propto f(\mathbf{x}_t|\Theta)\, f(\Theta|x(t-1)), \tag{4}$$

where $x(t)$ denotes a collection of all available data $\mathbf{x}_t$ up to the time instant t, i.e., $x(t) = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t\}$, including the prior data $\mathbf{x}_0$, and where $f(\Theta|x(t-1))$ denotes the prior pdf. The estimation approach recalled below can be found in [18, 34]. According to them, the model (2) is rewritten in the form

$$f(\mathbf{x}_t|\Theta) = (2\pi)^{-N/2}|\mathbf{R}|^{-1/2} \exp\left\{-\frac{1}{2}\mathrm{tr}\left(\mathbf{R}^{-1}\begin{bmatrix} -1 \\ \theta \end{bmatrix}'\mathbf{D}_t\begin{bmatrix} -1 \\ \theta \end{bmatrix}\right)\right\}, \tag{5}$$

where tr is a trace of the matrix and

$$\mathbf{D}_t = \begin{bmatrix} \mathbf{x}_t \\ 1 \end{bmatrix}\begin{bmatrix} \mathbf{x}_t \\ 1 \end{bmatrix}' \tag{6}$$

is the data matrix at time t. The posterior pdf $f(\Theta|x(t))$ in (4) has the conjugate prior Gauss-inverse-Wishart pdf with two recomputable statistics, which are respectively the information matrix $\mathbf{V}_{t-1}$ of the dimension $((N+1)\times(N+1))$ and the counter of the used data items κ_{t-1}, i.e.,

$$f(\Theta|x(t-1)) \propto |\mathbf{R}|^{-0.5\kappa_{t-1}} \exp\left\{-\frac{1}{2}\mathrm{tr}\left(\mathbf{R}^{-1}\left[\begin{array}{c}-1\\ \theta\end{array}\right]'\mathbf{V}_{t-1}\left[\begin{array}{c}-1\\ \theta\end{array}\right]\right)\right\}. \tag{7}$$

After substituting (5) and (7) into (4), the statistics are recursively recomputed as follows:

$$\begin{aligned} \text{the information matrix }\ \mathbf{V}_t &= \mathbf{V}_{t-1}+\left[\begin{array}{c}\mathbf{x}_t\\ 1\end{array}\right]\left[\begin{array}{c}\mathbf{x}_t\\ 1\end{array}\right]', & (8)\\ \text{counter }\ \kappa_t &= \kappa_{t-1}+1, & (9)\end{aligned}$$

where the initial statistics $\mathbf{V}_0$ and κ_0 start the recursion. They can be either (i) chosen with the zero or small values or (ii) computed from the available prior data or (iii) chosen by experts.

To obtain the point estimates of the parameters θ and $\mathbf{R}$ (not forgetting that $\Theta \equiv \{\theta,\mathbf{R}\}$), the updated information matrix is partitioned

$$\mathbf{V}_t = \left[\begin{array}{cc}\mathbf{V}_{xx} & V_x'\\ V_x & V_1\end{array}\right], \tag{10}$$

where $\mathbf{V}_{xx}$ is square matrix of dimension $(N\times N)$, V_x' is the N-dimensional column vector and V_1 is scalar. The point estimates at time t are computed as follows:

$$\hat{\theta}_t = V_1^{-1}V_x,\quad \hat{\mathbf{R}}_t = \frac{\mathbf{V}_{xx}-V_x'V_1^{-1}V_x}{\kappa_t}. \tag{11}$$

Further details can be found in [18, 34].

The approach for the estimation of the individual model (3) recalled below is available in [17], which proposes to use the conjugate prior Dirichlet pdf with the recursively updated statistics. The posterior pdf of the parameter α is evolved similarly to (4) with the corresponding data in the condition, i.e.,

$$f(\alpha|y(t)) \propto f\left(y_t=i|\alpha\right) f(\alpha|y(t-1)). \tag{12}$$

The recomputable statistics denoted by γ_t has a dimension of the transition table (3). It means that here it is a vector of the dimension K. Using a similar scheme, i.e., substituting the model (3) and the prior Dirichlet pdf into (12), the update of the statistics is done for the current value $y_t=i$ with $i\in\{1,2,\ldots,K\}$ as follows:

$$\gamma_{i;t} = \gamma_{i;t-1}+1, \tag{13}$$

where $\gamma_{i;t}$ are the entries of the statistics γ_t, and the initial statistics γ_0 (that can be either chosen as zero or computed from prior data or set by experts) starts the recursion. In practice it means that the statistics counts occurrences of values of y_t (notice that y_t should be measured for such an update). The point estimate of the parameter α is obtained by normalizing the statistics γ_t

$$\hat{\alpha}_{i;t} = \frac{\gamma_{i;t}}{\sum_{k=1}^{K}\gamma_{k;t}},\ i\in\{1,2,\ldots,K\}. \tag{14}$$

4. Multinomial mixture-based logistic regression

Formulating the problem of the multinomial logistic regression as the estimation of the pointer of the mixture model, it is necessary to take into account that in reality the straightforward update (13) is often not possible because of unavailable measurements of y_t. Two different situations can occur in the considered context. The first one is when values of y_t were measured during a limited period of time or are available in the form of prior (or simulated) data, however they are no longer observed. In the case of the informative data, they can be enough for estimating the pointer model (3) and subsequently for classifying the data $\mathbf{x}_t$. The solution to this situation can be strictly divided into the learning and the testing phases as stated in Section 2.

The second situation occurs when the pointer model should be periodically learned from the newly arriving data also during the testing phase. This can happen, for instance, when the values of y_t are measured with a longer period than $\mathbf{x}_t$, or they are suddenly missing due to measuring failures. In this case the learning and the testing phases are joined together. Solutions to both the situations via the mixture estimation based on [18,38] are presented below.

4.1 On-line/Off-line mixture-based logistic regression

The combination of the on-line and off-line logistic regression via the mixture estimation takes the following form. The idea is that firstly the models (1) and (3) are learned using the measured values of y_t and $\mathbf{x}_t$. Then the learned models are used for predicting the value of y_t and classifying the new data $\mathbf{x}_t$.

4.1.1 The learning phase

The derivation is based on construction of the joint pdf of all variables to be estimated and application of the Bayes rule and the chain rule [12,19,34]. The idea of estimating the mixture with the known active component [17,18,38] is used.

During the learning phase until the time $t = T$ the parameters Θ and α have to be estimated based on the available data collection up to the time $t = T$. The data collection is represented by $\{y(t), x(t)\}$ denoted by $\Delta(t)$, see notations in Section 3.2. The joint pdf is constructed as follows:

$$f(\Theta,\alpha|\Delta(t)) \propto \underbrace{f(\mathbf{x}_t, y_t = i, \Theta, \alpha|\Delta(t-1))}_{\text{joint pdf via Bayes rule}}$$

$$= \underbrace{f(\mathbf{x}_t|y_t = i, \Theta, \alpha, \Delta\,(t-1))\, f(y_t = i|\Theta, \alpha, \Delta\,(t-1))\, f(\Theta|\alpha, \Delta\,(t-1))\, f(\alpha|\Delta\,(t-1))}_{\text{joint pdf via the chain rule}}$$

$$= \underbrace{f(\mathbf{x}_t|\Theta_i)\, f(y_t = i|\alpha)\, f\left(\Theta|\Delta\,(t-1)\right) f\left(\alpha|\Delta\,(t-1)\right)}_{\text{by the independence assumptions}},\ i \in \{1, 2, \ldots, K\}\,, \tag{15}$$

where the following <u>independence assumptions</u> hold $\forall i \in \{1, 2, \ldots, K\}$:

$$f\left(\mathbf{x}_t|y_t = i, \Theta, \alpha, \Delta\,(t-1)\right) = f\left(\mathbf{x}_t|y_t = i, \Theta\right), \tag{16}$$

which is the model (1) for the active i-th component, i.e., $f\left(\mathbf{x}_t|\Theta_i\right)$,

$$f\left(y_t = i|\Theta, \alpha, \Delta\left(t-1\right)\right) = f\left(y_t = i|\alpha\right) \tag{17}$$

is the model (3) assumed not to be dependent on the old data up to the time $t-1$,

$$f\left(\Theta|\alpha, \Delta\left(t-1\right)\right) = f\left(\Theta|\Delta\left(t-1\right)\right) \tag{18}$$

is the prior pdf for estimating the parameter Θ, and $f\left(\alpha|\Delta\left(t-1\right)\right)$ is the prior pdf for estimating the parameter α, which means that parameters Θ and α are assumed to be mutually independent.

After grouping the pdfs in (15) between those connected to the components and the pointer, the righthand side of (15) takes the form

$$\underbrace{f\left(\mathbf{x}_t|\Theta_i\right) f(\Theta|\Delta(t-1)}_{\text{update via (4) for the active } i\text{-th component}} \quad \times \quad \underbrace{f\left(y_t = i|\alpha\right) f(\alpha|\Delta(t-1)),}_{\text{update via (12) for the current } y_t} \tag{19}$$

$i \in \{1, 2, \ldots, K\}$, which allows updating the statistics for the parameter estimation directly due to the measured values of y_t. The expression $f\left(\mathbf{x}_t|\Theta_i\right) f(\Theta|\Delta(t-1)$ in (19) represents (4) applied for the value $y_t = i$, available at the time instant t. Thus the statistics update (8)–(9) for the estimation of Θ is performed $\forall i \in \{1, 2, \ldots, K\}$ according to [18, 38] as follows:

$$\begin{aligned} \mathbf{V}_{i;t} &= \mathbf{V}_{i;t-1} + \delta(i, y_t) \begin{bmatrix} \mathbf{x}_t \\ 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_t \\ 1 \end{bmatrix}', && (20) \\ \kappa_{i;t} &= \kappa_{i;t-1} + \delta(i, y_t), && (21) \end{aligned}$$

where $\mathbf{V}_{i;t}$ and $\kappa_{i;t}$ denotes the statistics of the i-th component (see Section 3.2), and δ is the Kronecker delta function such that $\delta\left(i, y_t\right) = 1$ if the value i of the variable y_t has been measured at the time instant t, otherwise $\delta\left(i, y_t\right) = 0$.

The expression $f\left(y_t = i|\alpha\right) f(\alpha|\Delta(t-1))$ in (19) is the direct application of (12), since values of y_t are available. Thus the update of statistics for estimating the parameter α is done directly using (13) $\forall i \in \{1, 2, \ldots, K\}$ according to [17]:

$$\gamma_{i;t} = \gamma_{i;t-1} + \delta(i, y_t). \tag{22}$$

Since the data $\Delta(t)$ are available until $t = T$, the statistics stop to be actualized at the time instant $t = T$, which means the end of the learning phase. The point estimates of the parameter Θ are computed for each $i \in \{1, 2, \ldots, K\}$ according to (10)–(11) resulting in $\{\hat{\theta}_{i;t}, \hat{\mathbf{R}}_{i;t}\} \equiv \hat{\Theta}_{i;t}$. The point estimate of the parameter α is obtained using (14). Generally all point estimates can be computed at each time instant, but it is enough to obtain them at $t = T$ with the completely updated statistics. In this way in the end of the learning phase the learned models (1) and (3) are obtained.

4.1.2 The testing phase

During the testing phase for the time instants $t > T$ the pointer y_t is no longer measured. It means that the data collection $\Delta(t)$ includes now $\{\mathbf{x}_t, \Delta(T)\}$. Thus, the values of y_t should be recursively estimated for each $t > T$ and then used for classifying the new data $\mathbf{x}_t$ with the help of the learned models (1) and (3). Similarly to the learning phase, the derivation is based on the construction of the joint pdf of variables to be estimated and the Bayes and the chain rule using the mixture estimation approach proposed in [18].

The joint pdf now includes also the variable y_t, i.e., $\forall i \in \{1, 2, \ldots, K\}$

$$f(\Theta, y_t = i, \alpha|\mathbf{x}_t, \Delta(T)) \propto \underbrace{f(\mathbf{x}_t, \Theta, y_t = i, \alpha|\Delta(T))}_{\text{joint pdf via Bayes rule}}$$

$$= \underbrace{f\left(\mathbf{x}_t|\Theta, y_t = i\right) f(\Theta|\Delta(T) \times f\left(y_t = i|\alpha\right) f(\alpha|\Delta(T))}_{\text{via the chain rule and independence assumptions}}, \qquad (23)$$

where the last result is obtained similarly to (15) using the chain rule and the independence assumptions (16)–(18), taking into account that the model $f\left(\mathbf{x}_t|\Theta, y_t = i\right)$ instead of (1) should be used now because of the absence of the value of y_t.

To obtain the pdf for y_t, the result (23) should be marginalized over parameters Θ and α

$$f(y_t = i|\mathbf{x}_t, \Delta(T)) \propto \int_{\Theta^*}\int_{\alpha^*} \underbrace{f\left(\mathbf{x}_t|\Theta, y_t = i\right) f(\Theta|\Delta(T) \times f\left(y_t = i|\alpha\right) f(\alpha|\Delta(T))}_{(23)} d\Theta d\alpha$$

$$= \int_{\Theta^*} f\left(\mathbf{x}_t|\Theta, y_t = i\right) f(\Theta|\Delta(T)\, d\Theta \times \int_{\alpha^*} f\left(y_t = i|\alpha\right) f(\alpha|\Delta(T))\, d\alpha. \qquad (24)$$

The first integral in (24) is approximated using the Dirac delta function $\delta\left(\Theta, \hat{\Theta}_T\right)$ as the prior pdf of the parameter Θ, i.e., $\forall i \in \{1, 2, \ldots, K\}$

$$\int_{\Theta^*} f\left(\mathbf{x}_t|\Theta, y_t = i\right) f(\Theta|\Delta(T)\, d\Theta \doteq f\left(\mathbf{x}_t|\hat{\Theta}_{i;T}\right), \qquad (25)$$

which means that the point estimates $\hat{\Theta}_{i;T}$ obtained in the end of the learning phase are substituted into the corresponding i-th components. This approximation simplifies computations and offers a very good interpretation to the expression. The result of (25) provides the proximity of the current data $\mathbf{x}_t$ to each component. It is denoted by $\mathcal{L}_{i;x_t}$ and here it has the form $\forall i \in \{1, 2, \ldots, K\}$

$$\mathcal{L}_{i;x_t} = (2\pi)^{-N/2}|\hat{\mathbf{R}}_{i;T}|^{-1/2} \exp\left\{-\frac{1}{2}[\mathbf{x}_t - \hat{\theta}_{i;T}]'\hat{\mathbf{R}}_{i;T}^{-1}[\mathbf{x}_t - \hat{\theta}_{i;T}]\right\}. \qquad (26)$$

The second integral in (24) provides the point estimate of the parameter α using its statistics γ_T from the time instant $t = T$ according to (14).

After computing these two integrals the required pdf (24) takes the following form, using [18], $\forall i \in \{1, 2, \ldots, K\}$:

$$f(y_t = i|\mathbf{x}_t, \Delta(T)) \propto \quad \underbrace{\mathcal{L}_{i;x_t}\hat{\alpha}_{i;T},}_{\text{denoted by } \tilde{\mathrm{w}}_{i;t}} \qquad (27)$$

which is normalized

$$\mathrm{w}_{i;t} = \frac{\tilde{\mathrm{w}}_{i;t}}{\sum_{k=1}^{K} \tilde{\mathrm{w}}_{k;t}}, \tag{28}$$

where the values $\mathrm{w}_{i;t}$ are the probabilities of activity of the i-th component at time t. The obtained probabilities create the weighting vector $\mathbf{w}_t = [\mathrm{w}_{1;t}, \mathrm{w}_{2;t}, \ldots, \mathrm{w}_{K;t}]'$. During the testing phase for the time $t > T$ this weighting vector is enough for the task of classification of the data $\mathbf{x}_t$. They are classified as belonging to the component, corresponding to the biggest entry of the vector $\mathbf{w}_t$, which is the point estimate of the pointer y_t at time t.

Both the phases are summarized in the form of the following structural algorithm.

Algorithm 1

```
{Initialization, i.e., $t = 1$}
Specify $K$ components (2).
for all $i \in \{1, 2, \ldots, K\}$ do
  Set the initial values of the statistics $\mathbf{V}_{i;t}$, $\kappa_{i;t}$ for each component (2) and $\gamma_{i;t}$
  for the model (3). {See explanations in Section 3.2.}
end for
{The off-line learning phase}
for $t = 2, \ldots, T$ do
  Measure the data $\mathbf{x}_t$ and $y_t$.
  Update the active component, i.e., the statistics $\mathbf{V}_{i;t}$, $\kappa_{i;t}$ and $\gamma_{i;t}$ according
  to (20), (21) and (22) respectively.
  if $t = T$ then
    for all $i \in \{1, 2, \ldots, K\}$ do
      Compute the point estimates $\hat{\Theta}_{i;t} = \left\{\hat{\theta}_{i;t}, \hat{\mathbf{R}}_{i;t}\right\}$ and $\hat{\alpha}_{i;t}$ according to (11)
      and (14).
    end for
  end if
end for
{The on-line testing phase}
for $t = T + 1, T + 2, \ldots$ do
  Measure the new data $\mathbf{x}_t$.
  for all $i \in \{1, 2, \ldots, K\}$ do
    Compute proximities of the data $\mathbf{x}_t$ to each component via (26).
    Compute probabilities $\mathrm{w}_{i;t}$ for the weighting vector $\mathbf{w}_t$ according to (27)
    and (28).
  end for
  Classify the data $\mathbf{x}_t$ as belonging to the component, corresponding to the
  biggest entry of the vector $\mathbf{w}_t$. {This biggest entry is the point estimate of the pointer
  $y_t$ at time $t$. }
end for
```

4.2 On-line multinomial mixture-based logistic regression

The previous section provides the solution to the problem formulated in Section 2. This section adjusts it to the situation when it is not possible to accumulate the statistics from measured data y_t during the separate learning phase (for instance, because of measuring failures, a longer period of measuring $\mathbf{x}_t$ and y_t, etc.). In this case it is advantageous to join the learning and the testing phases, estimate parameters on-line only from the data $\mathbf{x}_t$ and use the predicted value of y_t for classification. However, if the observed values of y_t are available at the time instant t, they are used for learning instead its estimate. It leads to a combination of the previous algorithm with the mixture estimation from [18].

The derivations are identical to (23)–(28) used for the testing phase of the previous section except for the restriction by the time T. The data collection $\Delta(t)$ includes $x(t)$ always and y_t periodically with the time $t = 1, 2, \ldots$. It is not known before whether the values of y_t are measured or not, therefore y_t is among the variables to be estimated. Here the resulted pointer pdf has the form

$$f(y_t = i|\mathbf{x}_t, \Delta(t)) \propto \int_{\Theta^*} f\left(\mathbf{x}_t|\Theta, y_t = i\right) f(\Theta|\Delta(t)\, d\Theta \times \int_{\alpha^*} f\left(y_t = i|\alpha\right) f(\alpha|\Delta(t))\, d\alpha$$

$$\propto \quad \underbrace{\mathcal{L}_{i;x_t}\hat{\alpha}_{i;t-1}}_{\text{denoted by } \tilde{\mathrm{w}}_{i;t}} \quad \text{to be normalized via (28),} \qquad (29)$$

where the proximity $\mathcal{L}_{i;x_t}$ is obtained according to (26) with the previous point estimates $\hat{\theta}_{i;t-1}$ and $\hat{\mathbf{R}}_{i;t-1}$.

In (29) the point estimate $\hat{\theta}_{i;t-1}$, $\hat{\mathbf{R}}_{i;t-1}$ and $\hat{\alpha}_{i;t-1}$ are obtained at the previous time instant $t-1$ either using the direct updates (20), (21) and (22) respectively (if the value of y_t is observed), or with the help of the mixture estimation algorithm from [18] as follows. The statistics updates (20), (21) and (22) are performed with replacing the Kronecker function $\delta\left(i, y_t\right)$ by the probability $\mathrm{w}_{i;t}$ for each $i \in \{1, 2, \ldots, K\}$ obtained from (28) for the corresponding time instant, i.e.,

$$\mathbf{V}_{i;t} = \mathbf{V}_{i;t-1} + \mathrm{w}_{i;t}\begin{bmatrix} \mathbf{x}_t \\ 1 \end{bmatrix}\begin{bmatrix} \mathbf{x}_t \\ 1 \end{bmatrix}', \qquad (30)$$

$$\kappa_{i;t} = \kappa_{i;t-1} + \mathrm{w}_{i;t}, \qquad (31)$$

$$\gamma_{i;t} = \gamma_{i;t-1} + \mathrm{w}_{i;t}. \qquad (32)$$

The updated statistics are then used for recomputing the point estimates (11) and (14) for each $i \in \{1, 2, \ldots, K\}$. The structural algorithm is provided below.

5. Results

5.1 Example with simulated data

Let's demonstrate Algorithm 1 with the help of a simple example with simulated data. The discrete dependent variable y_t has 5 possible values, i.e., $y_t \in \{1, 2, 3, 4, K = 5\}$, and y_t is generated by the random generator from the uniform distribution. The data vector $\mathbf{x}_t = [\mathrm{x}_{1;t}, \mathrm{x}_{2;t}, \mathrm{x}_{3;t}, \mathrm{x}_{4;t}]'$ has four continuous entries.

Algorithm 2

```
{Initialization, i.e., $t = 1$}
  Specify $K$ components (2).
  for all $i \in \{1, 2, \dots, K\}$ do
    Set the initial values of the statistics $\mathbf{V}_{i;t}$, $\kappa_{i;t}$ for each component (2) and $\gamma_{i;t}$ for the model (3). {See explanations in Section 3.2.}
    Using these initial statistics, compute the point estimates $\hat{\Theta}_{i;t} = \left\{\hat{\theta}_{i;t}, \hat{\mathbf{R}}_{i;t}\right\}$ and $\hat{\alpha}_{i;t}$ according to (11) and (14).
  end for
{On-line learning and testing}
  for $t = 2, 3, \dots$ do
    Measure the data $\mathbf{x}_t$.
    if the data item $y_t$ is measured then
      Update the active component, i.e., the statistics $\mathbf{V}_{i;t}$, $\kappa_{i;t}$ and $\gamma_{i;t}$ according to (20), (21) and (22) respectively.

      for all $i \in \{1, 2, \dots, K\}$ do
        Recompute the point estimates $\hat{\Theta}_{i;t} = \left\{\hat{\theta}_{i;t}, \hat{\mathbf{R}}_{i;t}\right\}$ and $\hat{\alpha}_{i;t}$ according to (11) and (14).
      end for
    end if
    for all $i \in \{1, 2, \dots, K\}$ do
      Compute proximities (26) of the data $\mathbf{x}_t$ to each component, using $\hat{\theta}_{i;t-1}$ and $\hat{\mathbf{R}}_{i;t-1}$.
      Compute probabilities $\mathrm{w}_{i;t}$ according to (27) and (28) using $\hat{\alpha}_{i;t-1}$.
    end for
    Classify the data $\mathbf{x}_t$ as belonging to the component, corresponding to the biggest entry of the vector $\mathbf{w}_t$, which is the point estimate of the pointer $y_t$ at time $t$.
    for all $i \in \{1, 2, \dots, K\}$ do
      Update the statistics using (30)–(32).
      Recompute the point estimates $\hat{\theta}_{i;t-1}$, $\hat{\mathbf{R}}_{i;t-1}$ and $\hat{\alpha}_{i;t-1}$ according to (11) and (14) and go to the first step of the on-line part of the algorithm.
    end for
  end for
```

Simulation

The data $\mathbf{x}_t$ are generated for each value of y_t using the model (2) with the following parameters

$$\theta_1 = \begin{bmatrix} 3 \\ -1 \\ 5 \\ 4 \end{bmatrix}, \quad \theta_2 = \begin{bmatrix} -8 \\ 8 \\ -2 \\ 5 \end{bmatrix}, \quad \theta_3 = \begin{bmatrix} 13 \\ -10 \\ 0 \\ 6 \end{bmatrix}, \quad \theta_4 = \begin{bmatrix} 9 \\ 0 \\ 15 \\ 7 \end{bmatrix}, \quad \theta_5 = \begin{bmatrix} 0 \\ -8 \\ 10 \\ 8 \end{bmatrix}, \tag{33}$$

and with the same variances 3 placed in the diagonal of the covariance matrices $\mathbf{R}_i$, $\forall i \in \{1,2,3,4,5\}$.

Initialization

Since $K=5$, the initial statistics for 5 components are constructed as follows. The information matrix of each component, i.e., $\mathbf{V}_{i;t}$ $\forall i \in \{1,2,\ldots,K\}$ is chosen as the zero square matrix of the dimension $(N+1)=5$, where $N=4$ is the dimension of the vector $\mathbf{x}_t$. The counter κ_t is chosen as the zero vector of the dimension $K=5$. Each its entry corresponds to $\kappa_{i;t}$ of the i-th component. The initial statistics of the switching model γ_t is also chosen as the zero 5-dimensional vector.

The learning phase

The measured data y_t and $\mathbf{x}_t$ for $t=1,2,\ldots,T=300$ are used for the learning phase. Using these data the above statistics are updated for the value i equal to the actual measured y_t (i.e., for the active component) according to (20), (21) and (22) respectively, i.e.,

$$\mathbf{V}_{i;t} = \mathbf{V}_{i;t-1} + \begin{bmatrix} \mathrm{x}_{1;t} \\ \mathrm{x}_{2;t} \\ \mathrm{x}_{3;t} \\ \mathrm{x}_{4;t} \\ 1 \end{bmatrix} \begin{bmatrix} \mathrm{x}_{1;t} \\ \mathrm{x}_{2;t} \\ \mathrm{x}_{3;t} \\ \mathrm{x}_{4;t} \\ 1 \end{bmatrix}', \quad \kappa_{i;t} = \kappa_{i;t-1} + 1, \quad \gamma_{i;t} = \gamma_{i;t-1} + 1. \tag{34}$$

For the time instant $t=300$ the matrices $\mathbf{V}_{i;T}$ for each $i \in \{1,2,3,4,5\}$ are partitioned according to (10) so that $\mathbf{V}_{xx}$ is a matrix of dimension (4×4), V_x' is a 4-dimensional column vector and V_1 is a scalar. Then the point estimates of the regression coefficients are computed according to (11) as follows:

$$\hat{\theta}_{1;T} = \begin{bmatrix} 3.28 \\ -0.82 \\ 4.8 \\ 3.94 \end{bmatrix}, \hat{\theta}_{2;T} = \begin{bmatrix} -7.96 \\ 7.74 \\ -2.10 \\ 4.97 \end{bmatrix}, \hat{\theta}_{3;T} = \begin{bmatrix} 12.86 \\ -9.92 \\ -0.06 \\ 5.96 \end{bmatrix},$$

$$\hat{\theta}_{4;T} = \begin{bmatrix} 9.21 \\ -0.01 \\ 15.07 \\ 7.19 \end{bmatrix}, \hat{\theta}_{5;T} = \begin{bmatrix} -0.19 \\ -8.05 \\ 9.79 \\ 8.33 \end{bmatrix}. \tag{35}$$

Their values correspond to (33). The estimated covariance matrices are not shown here to save space, however their diagonal entries correspond to those used for the simulation. In the case of necessity of observing the estimation evolution, the point estimates could be computed during the estimation at each time instant and then plotted. The point estimate of the parameter of the switching model is computed according to (14) and obtained as

$$\hat{\alpha}_{i;T} = \begin{bmatrix} 0.190 & 0.205 & 0.200 & 0.197 & 0.208 \end{bmatrix} \tag{36}$$

that corresponds to the uniform distribution used for the simulation.

The testing phase

The testing phase lasts from $t = 301$ till $t = 1500$, when only the data $\mathbf{x}_t$ are measured. Each time instant from $t = 301$ till $t = 1500$ using the measured data item $\mathbf{x}_t$ the proximities (26) are computed for each $i \in \{1, 2, 3, 4, 5\}$. According to (26), (27) and (28) at each time instant the obtained point estimates are used for constructing the 5-dimensional weighting vector, where its each i-th entry is the probability of the current activity of the i-th component. The biggest entry of the weighting vector is the point estimate of y_t at each time instant during the testing phase. Average number of wrong estimates of y_t for 10 simulations with parameters (33) and various random generators is 2.97%.

The measured data vector $\mathbf{x}_t$ is classified as belonging to the active component indicated by this biggest entry at each time instant $t = 301, \ldots, 1500$ during the testing phase of the algorithm. It is difficult to show the detected clusters in the multidimensional space, that's why Fig. 1 demonstrates selected results by plotting two variables from the vector $\mathbf{x}_t$ against each other. A rest of results is of a similar quality.

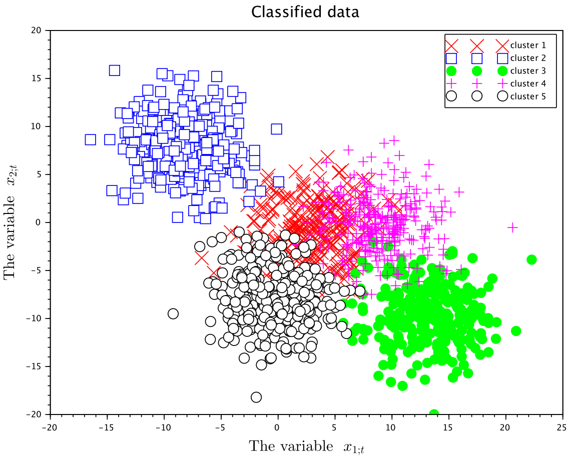

Fig. 1 *Selected classification results. The figure shows the variable $x_{1;t}$ plotted against $x_{2;t}$. Five clusters denoted by symbols $\times$, $\square$, $\bullet$, $+$, $\circ$ correspond to five possible values of the pointer y_t.*

The computation time of both the phases altogether is 0.5 seconds using the Scilab (see www.scilab.org) functions `tic` and `toc`.

In the case of using Algorithm 2, when the learning and the testing phase are joined, changes in this example are straightforward. The updates of statistics are done according to (30)–(32), when values of y_t are unavailable.

The aim of this example is to explain how the presented approach works and to verify the programming. Testing with real data is a much more challenging task. The results can be found in the next section.

5.2 Experiments with real data

Here the approach is tested on real data measured on a vehicle during driving. The main aim of the performed data analysis (not limited by that presented in this paper) is modeling the components characterizing different styles of driving. It can be, for instance, eco-, sporty and dangerous driving, a tired and drowsy driver, a sharply changed driving style that can signal sudden health problems of the driver or the fact that the car is driven by a different driver (e.g., it has been stolen), etc. A series of experiments was performed. Here the typical obtained results are demonstrated.

5.2.1 Data

The following measurements were selected for experiments.

The discrete dependent variable y_t expresses the gear selection during driving with the six-speed gearbox. It has the possible values: $\{-1, 0, 1, 2, 3, 4, 5, 6\}$, where 0 denotes the neutral gear and -1 is the reverse gear. The major part of the taken data represents driving with a relatively high speed out of city, which more corresponds to higher values of the gear. Because of this, values -1 and 1 were observed only rarely. To avoid their interpretation as outliers, values $\{-1, 0, 1\}$ were grouped into one value 1.

The explanatory variable $\mathbf{x}_t$ is the three-dimensional vector $[\mathrm{x}_{1;t}, \mathrm{x}_{2;t}, \mathrm{x}_{3;t}]'$, where $\mathrm{x}_{1;t}$ is the vehicle speed [km/h], $\mathrm{x}_{2;t}$ is position of the gas pedal [%], $\mathrm{x}_{3;t}$ is the engine speed [rpm].

5.2.2 Results

Here the results of application of both the algorithms are shown. Data measured each 0.2 seconds were taken for the experiments. The available data set was divided among 8 sets, each containing 9427 data items.

For Algorithm 1, a mix of data of the same size taken randomly over all data sets were used for the the learning phase, and 8 data sets for the testing phase. For Algorithm 2, these 8 data sets were tested so that 1000 (i.e., approximately one eighth) measured values of y_t were taken at random time instants during on-line running.

For comparison the following counterparts are chosen: (i) KNIME logistic regression tools (www.knime.org) and (ii) the Matlab functions `mnrfit` and `mnrval` (www.mathworks.com).

Tab. I demonstrates a percentage of incorrect estimates (PIE) of the discrete dependent variable y_t obtained by all the algorithms for 8 data sets. Columns of the table correspond to the data sets, rows to the compared methods.

The data used for the validation were measured in different traffic situations: from a relatively calm economic driving on the highway and a mixed driving on the first and second class roads to driving through several villages. This was done to obtain data covering as many driving styles as possible. Because of the character of data, the quality of estimation differs among the tested sets. Tab.I shows (compare, for example, results for data sets 1 and 4) that if PIE is higher, it grows for all the compared methods, and similarly it drops.

	1	2	3	4	5	6	7	8
Algorithm 1	2.10	3.50	10.93	12.22	9.04	8.72	6.78	7.00
Algorithm 2	3.21	3.24	6.04	11.79	5.95	10.98	4.93	5.66
KNIME tools	2.77	2.41	11.03	12.36	7.71	10.56	2.55	3.23
Matlab functions	4.95	4.63	11.66	12.36	10.23	15.52	7.97	6.85

Tab. I *Percentage of incorrect estimates for 8 tested data sets.*

Comparing PIE among the methods it can be seen that in general the results are close to each other. The average PIE of Algorithm 1 over 8 tested data sets is 7.54%. Algorithm 2 gives 6.48%, KNIME 6.58%, and Matlab functions 9.27%. It means that the most successful estimation was provided by Algorithm 2, and the worst results were with the Matlab functions. The difference is not too significant. However, closeness of the results to such trustful counterparts as the KNIME and Matlab estimators confirms that both Algorithms 1 and 2 can serve as an alternative logistic regression tool. A relatively successful estimation of real data of all the algorithms is also a part of the validation process.

Tab. II provides the average computation time (ACT) of the algorithms calculated by functions `tic` and `toc`. The computation time of Algorithms 1 and 2 is similar, and it is on the third place after the KNIME and Matlab estimators. But it should not be forgotten that both the Algorithms 1 and 2 work on the entirely different idea than the estimators used for the validation. They are running on-line (Algorithm 1 partially and Algorithm 2 completely) unlike to offline estimators of KNIME and Matlab, which means that further data could be classified if available.

	ACT, [s]
Algorithm 1	4.69
Algorithm 2	4.34
KNIME tools	2.11
Matlab functions	3.51

Tab. II *Average computation time of the compared algorithms.*

An example of graphic representation of results is shown in Fig. 2. A fragment of 7000 data items from the first tested data set was chosen for plotting. The most part of the data corresponds to driving on the highway with higher values of the gear y_t. These data were estimated similarly successfully by all the compared algorithms. However, the most interesting are fragments of the estimation with changing the gear values.

Fig. 2 (top) compares results of Algorithms 1 and 2 with real measurements. It can be seen that Algorithm 1 (which is at the third place with the average PIE) demonstrates several incorrect estimates around 2700, 3400, 5200 and 7500 data items. Algorithm 2, which has the least average PIE, covers changing the gear values successfully even around 8500 data items.

Fig. 2 (bottom) demonstrates results of the KNIME and Matlab estimators. In this fragment of estimation they are both mostly successful, excepting places near

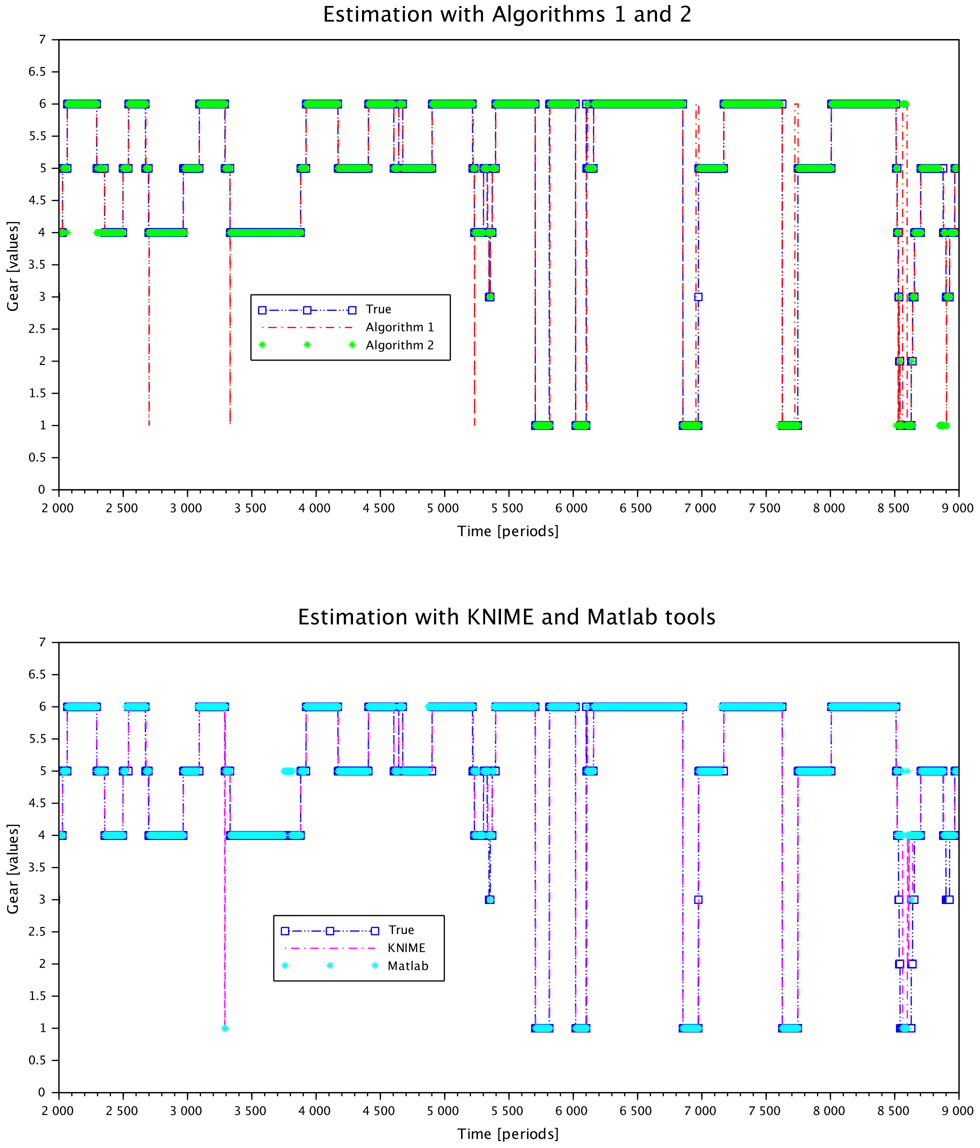

Fig. 2 *Comparison of results of estimating the gear values. The top figure provides results of Algorithms 1 and 2. The bottom figure shows results of the KNIME and Matlab estimators. Notice the difference of the results in all figures around* 2700*,* 3400*,* 5200*,* 7500 *and* 8500 *data items.*

3300, 3800 and 5400 data items, and they do not catch sudden changes of real gear values around 8500 data items. Visualization of the rest of tested data sets looks similar.

A colored visualization of classifying the data $\mathbf{x}_t$ requires too much space in the paper. To save space it is not shown here. The data are classified according to 6 estimated values of the gear among 6 clusters.

5.3 Discussion

To summarize the experimental part of the work, it should be noticed that the presented on-line alternative of the logistic regression was successfully validated with the help of 8 sets of real data. The results of both the algorithms are close to those provided by the taken counterparts, and sometimes even better. It confirms that the proposed idea of the mixture-based formulation of logistic regression is competitive. This can be decisive for application areas, requiring the real-time estimation of the active regime in which a considered system operates in dependence on continuous data.

6. Conclusions

The paper focuses on on-line modeling of discrete variables depending on continuous variables. The problem known as the logistic regression is considered via the recursive Bayesian estimation of mixture models. The main aim of the research is to explore a possibility of constructing the continuous data dependent switching model that should be estimated on-line. The formulated task requires avoiding numerical iterative methods. The paper demonstrates that the recursive mixture estimation theory in the presented interpretation can serve for solving the logistic regression problem. Several remarks to the discussed approach are given below:

- Due to a possibility to combine the learning and the testing phases either as two separate parts of the algorithms or as the one joint part, the approach can be tailored to a specific task depending on the availability of data.

- Existing solutions to the multinomial logistic regression are mostly based on the extension of the binary case. The presented approach does not make a difference about the number of possible values of the discrete dependent variable. The structure of the model is entirely different from that used in the logistic regression.

- The presented paper demonstrates the approach using the normal components. However, this is not a limitation of the approach. Different components with the reproducible statistics (state-space, exponential, categorical, etc.) can be used, which will allow to cover different types of data in dependence on specific tasks. In the case of the explanatory variable $\mathbf{x}_t$ of the mixed continuous and discrete nature the solution would require to add a discrete component (either static similar to (3) or the Markov model) for modeling a discrete entry of $\mathbf{x}_t$. Its statistics should be updated similarly to (13) according to [17], and the point estimates of its parameters should be obtained using (14). Since such a discrete entry is a measured variable, the change does not bring any computational complexity. This is planned to be published elsewhere.

- The provided validation experiments demonstrate promising results.

The approach also contributes to the systematic extension of the recursive mixture estimation algorithms published in [29, 30]. However, the open problems still remain, including e.g., the following:

- The estimation algorithms can be extended for different combinations of components and the switching model.
- Multi-step-ahead mixture prediction within the considered context is a separate task planned to be solved. It is expected that the dynamic mixture prediction algorithm will be a significant contribution in the field of classification-related problems.
- Extension of the switching model up to several delayed values in the regression vector is a further planned task.

Acknowledgement

This work was supported by projects GAČR GA15-03564S.

References

[1] ALLISON P.D. Convergence failures in logistic regression. In: *SAS global forum 2008, Statistics and data analysis*, 2008, 360.

[2] BIONDO S., RAMOS E., DEIROS M., RAGUÉ J.M., DE OCA J., MORENO P., FARRAN L., JAURRIETA E. Prognostic factors for mortality in left colonic peritonitis: a new scoring system. *Journal of the American College of Surgeons.* 2000, 191(6), pp. 635–642, doi: 10.1016/s1072-7515(00)00758-4.

[3] BOLDEA O., MAGNUS J.R. Maximum likelihood estimation of the multivariate normal mixture model. *Journal of The American Statistical Association.* 2009, 104(488), pp. 1539–1549, doi: 10.1198/jasa.2009.tm08273.

[4] BOLSTAD W. *Understanding computational Bayesian statistics.* Wiley, 2009, doi: 10.1002/9780470567371.

[5] CHEN R., LIU J.S. Mixture Kalman filters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology).* 2000, 62(3), pp. 493–508, doi: 10.1111/1467-9868.00246.

[6] CHIB S., JELIAZKOV I. Accept–reject Metropolis–Hastings sampling and marginal likelihood estimation. *Statistica Neerlandica.* 2005, 59(1), pp. 30–44, doi: 10.1111/j.1467-9574.2005.00277.x.

[7] CUESTA-ALBERTOS J.A., MATRÁN C., MAYO-ISCAR A. Robust estimation in the normal mixture model based on robust clustering. *Journal of the Royal Statistical Society: Series B (Statistical Methodology).* 2008, 70(4), pp. 779–802, doi: 10.1111/j.1467-9868.2008.00657.x.

[8] DUMOUCHEL W. Multivariate Bayesian logistic regression for analysis of clinical study safety issues. *Statistical Science.* 2012, 27(3), pp. 319–339, doi: 10.1214/11-sts381.

[9] FREEDMAN D.A. *Statistical Models: Theory and Practice.* Cambridge University Press, 2005, doi: 10.1017/CBO9781139165495.

[10] FROLOV A., HÚSEK D., POLYAKOV P. Y. , ŘEZANKOVÁ H. A comparative study of two methodologies for binary datasets analysis. *Neural Network World*. 2012, 22(6), pp. 565–582, doi: 10.14311/nnw.2012.22.035.

[11] FRÜHWIRTH-SCHNATTER S. *Finite Mixture and Markov Switching Models.* New York: Springer-Verlag, 2006, doi: 10.1007/978-0-387-35768-3.

[12] GELMAN A., CARLIN J.B., STERN H.S., DUNSON D.B., VEHTARI A., RUBIN D.B. *Bayesian Data Analysis (Chapman & Hall/CRC Texts in Statistical Science).* 3rd ed., Chapman and Hall/CRC, 2013.

[13] GELMAN A., JAKULIN A., PITTAU M.G., SU Y.-S. A weakly informative default prior distribution for logistic and other regression models. *The Annals of Applied Statistics.* 2008, 2(4), pp. 1360–1383, doi: 10.1214/08-aoas191.

[14] GHAHRAMANI Z., HINTON G.E. Variational learning for switching state-space models. *Neural Computation.* 2000, 12(4), pp. 831–864, doi: 10.1162/089976600300015619.

[15] GUPTA M.R. Theory and use of the EM method. *Foundations and Trends in Signal Processing.* 2010, 4(3), p. 223–296, doi: 10.1561/2000000034.

[16] HOSMER JR. D.W., LEMESHOW S., STURDIVANT R.X. *Applied Logistic Regression.* 3rd ed. Wiley, 2013, doi: 10.1002/9781118548387.

[17] KÁRNÝ M., BÖHM J., GUY T. V., JIRSA L., NAGY I., NEDOMA P., TESAŘ L. *Optimized Bayesian Dynamic Advising: Theory and Algorithms.* London: Springer-Verlag, 2006, doi: 10.1007/1-84628-254-3.

[18] KÁRNÝ M., KADLEC J., SUTANTO E.L. Quasi-Bayes estimation applied to normal mixture. In: ROJÍČEK J., VALEČKOVÁ M., KÁRNÝ M., WARWICK K., ed. *Preprints of the 3rd European IEEE Workshop on Computer-Intensive Methods in Control and Data Processing*, Prague, Czech Republic, 1998, pp. 77–82.

[19] KOCH K.-R. *Introduction to Bayesian Statistics.* 2nd ed. Springer, 2007, doi: 10.1007/978-3-540-72726-2.

[20] KOLOGLU M., ELKER D., ALTUN H, SAYEK I. Validation of MPI and OIA II in two different groups of patients with secondary peritonitis. *Hepato-Gastroenterology.* 2001, 48(37), pp. 147–151.

[21] LIN C.J. , WENG R.C., KEERTHI S.S. Trust region Newton method for logistic regression. *J. Mach. Learn. Res.*. 2008, 9, pp. 627–650.

[22] LIU C. Robit regression: a simple robust alternative to logistic and probit regression. In: GELMAN A., MENG X.-L., ed. *Applied Bayesian Modeling and Causal Inference from Incomplete-Data Perspectives: An Essential Journey with Donald Rubin's Statistical Family. John Wiley & Sons, Ltd, Chichester*, 2004, doi: 10.1002/0470090456.ch21.

[23] MARUYAMA Y., STRAWDERMAN W.E. A new Monte Carlo sampling in Bayesian probit regression. arXiv preprint, 2012. Available from: http://arxiv.org/pdf/1202.4339.pdf

[24] MATTOZO T.C., SOARES DA SILVA G., FERNANDES NETO A.P., COSTA J.A.F. Logistic regression applied to airport customer satisfaction using hierarchical quality model. In: *Lecture notes in computer science.* Springer Berlin Heidelberg, 2012, pp. 558–567, doi: 10.1007/978-3-642-32639-4_68.

[25] MCCORMICK T.M., RAFTERY A.E., MADIGAN D., BURD R.S. Dynamic logistic regression and dynamic model averaging for binary classification. *Biometrics.* 2012, 68(1), pp. 23–30, doi: 10.1111/j.1541-0420.2011.01645.x.

[26] MCGRORY C.A., TITTERINGTON D.M. Variational Bayesian analysis for hidden Markov models. *Australian & New Zealand Journal of Statistics.* 2009, 51, pp. 227–244, doi: 10.1111/j.1467-842x.2009.00543.x.

[27] MENARD S. *Applied logistic regression analysis.* 2nd ed. SAGE Publications Inc., 2002, doi: 10.4135/9781412983433.

[28] MINKA T. P. *Algorithms for maximum-likelihood logistic regression.* Carnegie Mellon University, 2003. Research report. Available from: http://repository.cmu.edu/cgi/viewcontent.cgi?article=1191&context=statistics

[29] NAGY I., SUZDALEVA E. Mixture estimation with state-space components and Markov model of switching. *Applied Mathematical Modelling.* 2013, 37(24), pp. 9970–9984, doi: 10.1016/j.apm.2013.05.038.

[30] NAGY I., SUZDALEVA E., KÁRNÝ M., MLYNÁŘOVÁ T. Bayesian estimation of dynamic finite mixtures. *Int. Journal of Adaptive Control and Signal Processing.* 2011, 25(9), pp. 765–787, doi: 10.1002/acs.1239.

[31] NG S.K., MCLACHLAN G.J. Mixture models for clustering multilevel growth trajectories. *Computational Statistics & Data Analysis.* 2014, 71, pp. 43–51, doi: 10.1016/j.csda.2012.12.007.

[32] NICOLAU J. An analysis of the 2002 presidential elections using logistic regression. *Brazilian political science review.* 2007, 1(1), pp. 125–135.

[33] PARK B.J., ZHANG Y., LORD D. Bayesian mixture modeling approach to account for heterogeneity in speed data. *Transportation Research Part B: Methodological.* 2010, 44(5), pp. 662–673, doi: 10.1016/j.trb.2010.02.004.

[34] PETERKA V. Bayesian Approach to System Identification. In: P. EYKHOFF, ed. *Trends and Progress in System Identification.* New York: Pergamon, 1981, pp. 239–304, doi: 10.1016/B978-0-08-025683-2.50013-2.

[35] POURHOSEINGHOLI M.A., MEHRABI Y., ALAVI-MAJD H., YAVARI P. Using latent variables in logistic regression to reduce multicollinearity, A case-control example: breast cancer risk factors. *Italian journal of public health.* 2008, 5(1), pp. 65–71.

[36] ŠMÍDL V., QUINN A. *The Variational Bayes Method in Signal Processing.* Springer-Verlag Berlin Heidelberg, 2006, doi: 10.1007/3-540-28820-1.

[37] THOMAS L.C., MUN JUNG K., THOMAS S.D., WU Y. Modeling consumer acceptance probabilities. *Expert Systems with Applications.* 2006, 30, pp. 499–506, doi: 10.1016/j.eswa.2005.10.011.

[38] TITTERINGTON D.M., SMITH A.F.M., MAKOV U.E. *Statistical Analysis of Finite Mixture Distributions (Wiley Series in Probability and Statistics – Applied Probability and Statistics Section).* Wiley, 1986.

[39] WATANABE K., KOBAYASHI T, OTSU N. Efficient optimization of logistic regression by direct use of conjugate gradient. In: *Proceedings of the 10th International Conference on Machine Learning and Applications and Workshops (ICMLA)*, Honolulu, HI. IEEE, 2011, pp. 496–500, doi: 10.1109/icmla.2011.63.

[40] XIAN WANG H., LUO B., BIN ZHANG Q., WEI S. Estimation for the number of components in a mixture model using stepwise split-and-merge EM algorithm. *Pattern Recognition Letters.* 2004, 25(16), pp. 1799–1809, doi: 10.1016/j.patrec.2004.07.007.

[41] XIONG Y., MANNERING F.L. The heterogeneous effects of guardian supervision on adolescent driver-injury severities: A finite-mixture random-parameters approach. *Transportation Research Part B: Methodological.* 2013, 49, pp. 39–54, doi: 10.1016/j.trb.2013.01.002.

[42] YU J. Fault detection using principal components-based Gaussian mixture model for semiconductor manufacturing processes. *IEEE Transactions on Semiconductor Manufacturing.* 2011, 24(3), pp. 432–444, doi: 10.1109/tsm.2011.2154850.

[43] YU J. A nonlinear kernel Gaussian mixture model based inferential monitoring approach for fault detection and diagnosis of chemical processes. *Chemical Engineering Science.* 2012, 68(1), pp. 506–519, doi: 10.1016/j.ces.2011.10.011.

[44] YU J. A particle filter driven dynamic Gaussian mixture model approach for complex process monitoring and fault diagnosis. *Journal of Process Control.* 2012, 22(4), pp. 778–788, doi: 10.1016/j.jprocont.2012.02.012.

[45] ZENG H., CHEUNG Y.-M. Learning a mixture model for clustering with the completed likelihood minimum message length criterion. *Pattern Recognition.* 2014, 47(5), pp. 2011–2030, doi: 10.1016/j.patcog.2013.09.036.

[46] ZOU Y., ZHANG Y., LORD D. Analyzing different functional forms of the varying weight parameter for finite mixture of negative binomial regression models. *Analytic Methods in Accident Research.* 2014, 1, pp. 39–52, doi: 10.1016/j.amar.2013.11.001.